UIDAI

Unique Identification Authority of India
Planning Commission, Govt. of India (GoI),
3rd Floor, Tower II,
Jeevan Bharati Building,
Connaught Circus,
New Delhi 110001

DOCUMENT NUMBER: UID _603_ECMP

AADHAR ENROLMENT CLIENT-REGISTRAR INTEGRATION MANUAL

PRE-ENROLMENT DATA, SECURITY & KYR+ DATA

DOCUMENT VERSION 2.2

ENROLMENT CLIENT APPLICATION VERSION 2.2

Table of Contents

1. Introduction

UIDAI was created with the responsibility to plan and implement a Unique Identification scheme, own and operate the CIDR and be responsible for updates and maintenance on an ongoing basis.

Since the biometrics capture using sophisticated biometric devices by highly trained professionals could be the bottle-neck in the entire enrolment processes, it is desirable to complete the other demographic data capture and verification ahead of the biometric capture. This step is called pre-enrolment. It is conceivable that through existing databases available to the Registrars (For example, state governments BPL, PDS, or NREGA databases) a pre-enrolment workflow can import data into the AADHAAR Enrolment Client module, and only call upon residents to capture biometric data so that it can be linked to the existing demographic record for that resident. Note that, even with pre-enrolment, final verification is done at the time of enrolment during which biometric data will also be captured.

Pre-enrolment data should be in a CSV file format as prescribed in this document. AADHAAR Enrolment Client software collects basic AADHAAR fields which are known as KYR (Know Your Resident) fields where as Registrar collects additional fields known as KYR+. AADHAAR Enrolment Client software provides a loosely coupled way to integrate KYR data into Registrar's software.

At the end of every enrolment, Enrolment Client software writes KYR fields into a pre-configured directory as per software manual as a name- value pair file which Registrar's software can load and continue to collect additional KYR+ fields. This document also lists the name- value pair file format required for the Registrars for launching their KYR+ application. The allowed data format and the allowed character length have been covered in the subsequent sections of this document.

The document covers the data format and the data length (maximum) supported with various labels. The registrars are expected to input the data in the suggested data format and length.

The document covers the registrar security integration. The registrar is expected to share the public key in the specified format with which the client application will encrypt their set of packets. The same set of packets needs to be decrypted at the registrar's end for their internal purpose. The document elaborates the decryption mechanism which the registrar needs to adhere to for successful packet decryption.

1.1 Terminology

Enrolment: is the process of capturing resident data (including demographic and biometric data). The enrolment is done by the enrolment operator/supervisor/agencies.

Enrolment centre: is the location where the enrolment happens. Each enrolment station has the required enrolment set-up to make the enrolment possible.

Enrolment station: is the system which does the enrolment capture. The enrolment set-up includes a computer, the biometric devices and some accessories.

Resident: is a person who undergoes the enrolment capture process and gets a UID from the government.

AADHAAR Number: is a 12-digit number issued by the government as proof of identity and residence in India.

Central ID Data Repository (CIDR): a repository regulated and managed by the UIDAI. It issues AADHAAR numbers, updates resident information and authenticates the identity of residents as required.

1.2 Objective of this document

The objective of this integration manual is to educate the intended audience, the Registrar to integrate the pre-enrolment data with the AADHAAR Enrolment Client application as well as to integrate the KYR+ related fields to KYR+ application. This document also helps the registrar in the successful decryption of the packets.

The objective of this document is not to cover the installation and configuration of the AADHAAR Enrolment Client. For the intended users of this document, the AADHAAR Enrolment Client application should be seamless and hence the user didn't bother about the internal demarcations of the system.

1.3 Abbreviations and Glossary

Abbreviation	**Full Form**
AES	Advanced Encryption Standard
CSV	Comma Separated File
HMAC	Hash-based Message Authentication Code
OAEP	Optimal Asymmetric Encryption Padding
PKI	Public Key Infrastructure
RSA	Algorithm designed by Rivest, Shamir and Adleman.
UIDAI	Unique Identification Authority of India

2. Integration Process

This section covers how to integrate the pre-enrolment data to the AADHAAR Enrolment Client. This document well illustrates the integration of the KYR+ data fields required for the registrar's application from the Enrolment Client.

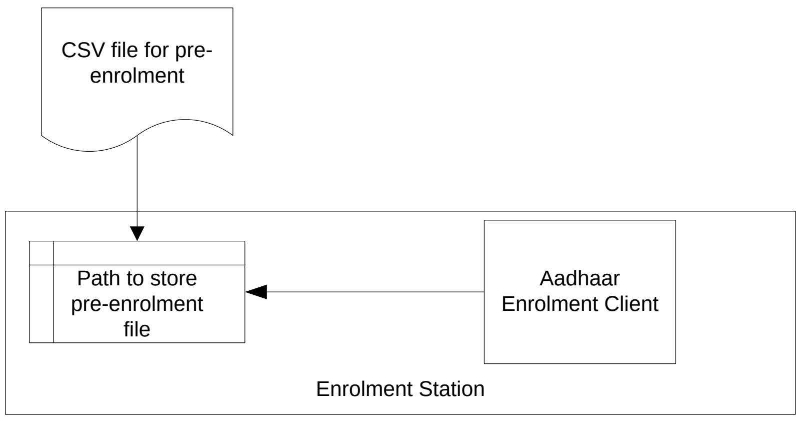
The subsequent sections cover the file templates required for the Registrar as well the KYR+ application. The fields/labels listed in the table shall not necessary feature in the same order as mentioned in the file format. The file shall contain all the labels as defined and no column with that label should be missing from the file. The Registrar can pass on the other fields, the details of which are not a part of the template file. The Enrolment Client application is intelligent enough to pick the labels required for running the application. The fields defined as part of pre-enrolment file is adhere to the DDVSP committee report circulated by UIDAI.

The last section covers the decryption process of the registrar packets.

2.1 Pre-Enrolment Data Integration

Since the biometrics capture using sophisticated biometric devices by highly trained professionals could be the bottle-neck in the entire enrolment processes, it is desirable to complete the other demographic data capture and verification ahead of the biometric capture. This step is called pre-enrolment.

In the case where Registrar has a good database, it can be used to pre-populate the AADHAAR Enrolment Client. This data is required to reduce the effort and time of the enrolment operators during enrolment capture process.



The typical process to integrate the pre-enrolment data file with the AADHAAR Enrolment Client application is as follows:

- ❖ Prepare a CSV (Comma separated Value) file for the pre-enrolment data in the template suggested in the subsequent section
- ❖ Ensure that the CSV file is stored in the pre-defined location as suggested in the installation guide in the enrolment station

The CSV file will contain the headers and all these headers should feature in the same CSV file. The registrar should make sure that the labels are not missing from the CSV file though the field level value may or may not be available for that label.

The table has been broken down in two fields, mandatory and optional fields.

Mandatory	PreEnrolmentID
Optional	LocalLanguageCode
	FullName
	FullName_LL
	Gender
	DateOfBirth
	DOBType
	PinCode
	POBox
	POType
	POLocal
	CareOfSalutation
	AddrCareOf
	AddrCareOf_LL
	AddrBuilding
	AddrBuilding_LL
	AddrStreet
	AddrStreet_LL
	AddrLandmark
	AddrLandmark_LL
	AddrLocality
	AddrLocality_LL
	AddrVTC
	AddrVTC_LL
	AddrDistrict
	AddrDistrict_LL
	AddrState
	AddrState_LL
	Mobile
	Email
	POI
	POA
	IntroducerName
	IntroducerName_LL
	IntroducerUID
	ProofHOF

	RelationType
	RelativeName
	RelativeName_LL
	RelativeUID
	RelativeEnrolmentNO
	RelativeEnrolmentDate
	TinNumber
	BankName
	BankState
	BankBranch
	BranchIFSCCode
	AccountNumber
	ApplicationNumber
	PreEnrolOperatorID
	PreEnrolOperatorName

The labels which shall be part of the pre-enrolment CSV file are as explained as below:

- **PreEnrolmentID**- This is the document ID for the document which the resident has. For example, Ration card number, Passport number etc. This is a mandatory field.
- **LocalLanguageCode**- This is the local language code for all the Indian state official language. The registrar may select the appropriate language code in accordance to his data and should ensure that the details to be mentioned in all the labels suffixed with 'LL' should be in the same local language. This is a mandatory field. For example, 06-Hindi. The local language code shall be followed as per MDDS: 01 Version: 1.0 specification for the languages as per the ISO 639-3.[1]
- **FullName**- This is the Name of the resident in English. The FullName follow to the data type format and length as described in section 3 of this document.
- **FullName_LL**- This is the Name of the resident in the local language as per the local language code mentioned.
- **DateOfBirth**- This is the Date of Birth of the resident. This shall adhere to the data type format and length as described in section 3 of this document. This should be in DDMMYYY format.
- **Gender**- This is the Gender of the resident. This shall adhere to the data type format and length as described in section 3 of this document. For example, M-Male, F-Female, and T-Transgender.
- **AddrCareOf**- This is the 'Care of' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrCareOf_LL**- This is the 'Care of' detail in the local language as per the local language code mentioned.

[1] Refer document Metadata and Data Standards for Person Identification and Land Region Codification MDDS: 01 Version: 1.0 December 24, 2009

- **AddrBuilding**- This is the 'building' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrBuilding_LL**- This is the 'building' detail in the local language as per the local language code mentioned.
- **AddrStreet**- This is the 'Street' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrStreet_LL**- This is the 'Street' detail in the local language as per the local language code mentioned.
- **AddrLandmark**- This is the 'landmark' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrLandmark_LL**- This is the 'landmark' detail in the local language as per the local language code mentioned.
- **AddrLocality**- This is the 'locality' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrLocality_LL**- This is the 'locality' detail in the local language as per the local language code mentioned.
- **AddrVTC**- This is the 'Village/Town/City' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrVTC_LL**- This is the 'Village/Town/City' detail in the local language as per the local language code mentioned.
- **AddrDistrict**- This is the 'District' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrDistrict_LL**- This is the 'District' detail in the local language as per the local language code mentioned.
- **AddrState**- This is the 'State' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **AddrState_LL**- This is the 'State' detail in the local language as per the local language code mentioned.
- **PinCode**-this is the Pincode details of the address of the resident.
- **RelativeName**- This is the 'Relative Name' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **RelativeName_LL**- This is the 'Relative Name 'detail in the local language as per the local language code mentioned.
- **RelativeUID**- This is the 'Relative UID or Enrolment ID' detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **RelationType**- This is the 'Relative Name detail of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
    - F-Father
    - M-Mother
    - H-Husband
    - W-Wife
    - G-Guardian

- **Mobile**- This is the Mobile number of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **Email**- This is the Email address of resident in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **ApplicationNumber**- This is the application number printed on the paper form which the resident's fills for enrolment. This shall adhere to the data type format and length as described in section 3 of this document.
- **TinNumber-** This is the NPR receipt number provided by the RGI to the resident. This shall adhere to the data type format and length as described in section 3 of this document.
- **PoI-** This is the Proof of Identity document provided by the resident in English. This is as per the list of documents prescribed by UIDAI under DDSVP report. This shall adhere to the data type format and length as described in section 3 of this document.
- **PoA-** This is the Proof of Address document provided by the resident in English. This is as per the list of documents prescribed by UIDAI under DDSVP report. This shall adhere to the data type format and length as described in section 3 of this document.
- **IntroducerName-** This is the Name of the Introducer in English. This shall adhere to the data type format and length as described in section 3 of this document.
- **IntroducerName_LL-** This is the 'Introducer Name 'detail in the local language as per the local language code mentioned.
- **IntroducerUID-** This is the UID of the Introducer. This shall adhere to the data type format and length as described in section 3 of this document.
- **ProofHOF-** This is the Proof of Head of Family document provided by the resident. This is as per the list of documents prescribed by UIDAI. This shall adhere to the data type format and length as described in section 3 of this document.
- **Bank Name-** This is the name of the bank where the resident owns an account. This bank shall be a valid bank. This shall adhere to the data type format and length as described in section 3 of this document.
- **Bank Branch Name-** This is the branch where the resident has his account in that bank. This shall adhere to the data type format and length as described in section 3 of this document.
- **Bank Branch IFSC code**- This is the IFSC code of the branch. In case, the branch has no IFSC code, the IFSC code will not be available. This shall adhere to the data type format and length as described in section 3 of this document.
- **Account Number-** This is the bank account number of the resident. This should be a valid account number. This shall adhere to the data type format and length as described in section 3 of this document.
- **PreEnrolOperatorID-** This is the operator ID for the operator who has captured the pre-enrolment details of the resident.
- **PreEnrolOperatorName-** This is the Name for the operator in English who has captured the pre-enrolment details of the resident.

2.2 Registrar Security Integration

The AADHAAR Enrolment Client application captures the demographic and biometric data of the residents. All the data collected by UIDAI will be encrypted with a 1024 / 2048 bit public key. In order for the registrars to get access to the enrolment packets specific to registrars, it is essential for them to provide their public key to the UIDAI. The public key file should be digitally signed by the CA for validation. UIDAI would integrate the given public key with the enrolment client. A release would be made with the available key to the respective registrars.

UIDAI recommends using multiple public keys to strengthen the security of the resident data. The public keys should be provided by the registrar before the registrar on-boarding process. The public keys will be integrated into the enrolment client. If a public key is not provided by the registrar, the registrar packets would not be created. The accepted **Public Key Format** for the AADHAAR enrolment client application is **Class 3** Web based **X509 DER** binary encoded.

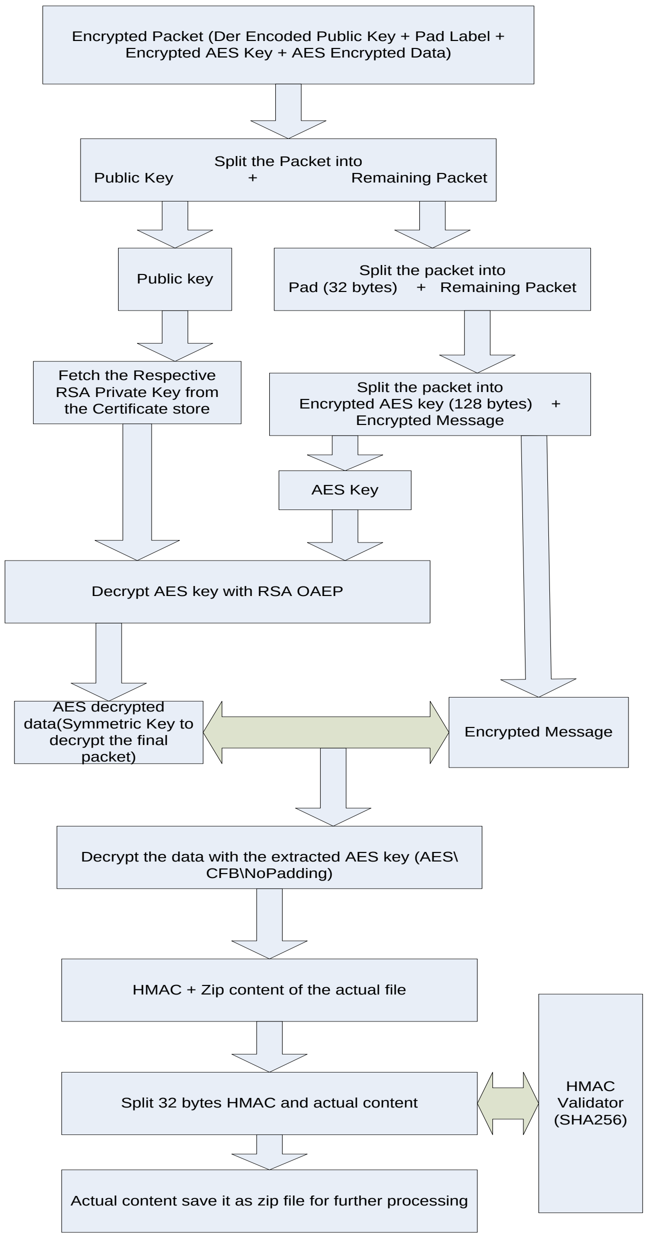
The Registrar code '000' to '010' are reserved for internal/training use. Hence, it is advised to not to use these codes as their registrar code.

The registrar packets which got created need to be successfully decrypted for internal purpose. The packets can be decrypted with the help of the public-private key combination at the registrar end.

2.2.1 Decryption of Enrollment Data Packets

The following figure depicts the flow for the decryption of the encrypted enrollment data packets. The section following the flow chart details the process followed in details.

The decryption logic uses the RSA-OAEP & AES encryption algorithms to decrypt the content. The entire byte stream is decrypted.


Encrypted Packet (Der Encoded Public Key + Pad Label + Encrypted AES Key + AES Encrypted Data)
Split the Packet into Public Key + Remaining Packet
Public key
Split the packet into Pad (32 bytes) + Remaining Packet
Fetch the Respective RSA Private Key from the Certificate store
Split the packet into Encrypted AES key (128 bytes) + Encrypted Message
AES Key
Decrypt AES key with RSA OAEP
AES decrypted data(Symmetric Key to decrypt the final packet)
Encrypted Message
Decrypt the data with the extracted AES key (AES\CFB\NoPadding)
HMAC + Zip content of the actual file
Split 32 bytes HMAC and actual content
HMAC Validator (SHA256)
Actual content save it as zip file for further processing

2.2.1.1 Packet Structure

The packet is structured as follows.

Content		Size
DER encoded Public Key		162 bytes
OAEP Pad		32 bytes
Encrypted AES Key (RSA-OAEP padding)		128 bytes
AES Encrypted data(the encrypted data comprises of HMAC + original zip file)		(all the remaining data)
HMAC	Encrypted data (Original zip file)	HMAC size 32 bytes

- **Encrypted Enrollment Packet:** DER Encoded Public Key + Padded Label + Encrypted AES Key + AES Encrypted Data
- **Extracting the Public key:** The RSA public key (162 bytes) is extracted from the packet and decoded. The keys are stored in the packet in DER format.
- **Decrypt the Private Key using master key**: The private key is obtained from the certificate store
- **Decrypting AES key with RSA OAEP:** The encrypted AES key is extracted from the encrypted packet. Then, AES symmetric key is decrypted back using RSA private key.
- Mode for Decryption: RSA/ECB/OAEPWITHSHA-256ANDMGF1PADDING
- **AES Decryption:** The encrypted data is decrypted using AES symmetric key.
- Mode for AES Decryption: AES/CFB/No Padding
- **Integrity checking**: The HMAC is extracted from the decrypted data. After removing original HMAC, for the remaining data the new HMAC is again calculated using SHA256. The 2 HMAC values are compared – if they are equal it shows the integrity is preserved else it means the enrolment data has been tampered.

2.2.1.2 Registrar De-cryptor Utility

This test de-cryptor utility is developed to assist Registrars in decrypting the Registrar Enrolment Packets before go-live. The utility supports only **PFX** and **PEM** formats for private key files. The private key files must be password protected. The decryptor tool and the source code is available at http://developer.uidai.gov.in/site/downloads

This is designed purely as a test tool along with source code provided for initial testing. THIS IS NOT SUPPORTED SOFTWARE.

Steps for using the utility –

1) Select the private key file (*.pfx, *.pem)
2) Enter the correct password
3) Select the Encrypted Registrar Packet
4) Please browse an appropriate location where registrar wants to save decrypted packet
5) Enter the name for output zip file. This is the original file which was encrypted for registrars!

Messages catalogue

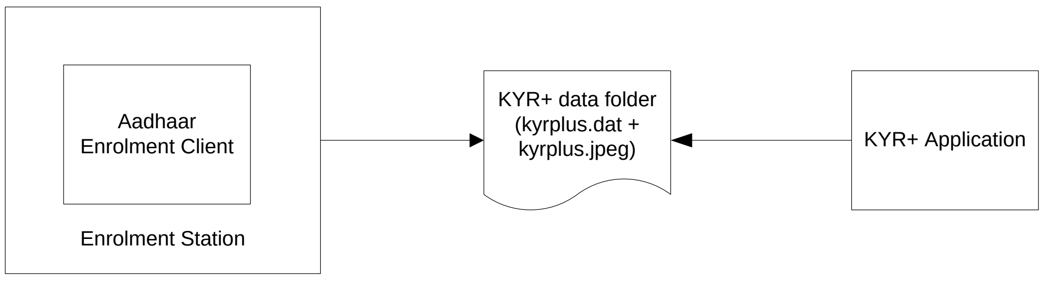
Error Message	**Description**
Failed : PKCS12 Key store MAC invalid – wrong password or corrupt file	Password provided for the private key file (.pfx or .pem) is wrong, or the file is corrupted. Therefore, the utility is not able to read the file.
Failed : data hash wrong	The private key is invalid
Failed : input too large for RSA cipher	
Failed: File is Corrupt or invalid!	The encrypted packet is either corrupted or tampered.
Successful!	The decryption is successful.

When the packets are successfully decrypted, the registrar shall be able to view the various files which contain the demographic details and the biometric details of the resident. The registrar can use these files to successfully integrate with their specific application.

2.3 KYR+ Data Integration

The AADHAAR Enrolment Client application captures the KYR (Know Your Resident) data. The registrars may require capturing some other registrar specific fields related to residents called as KYR+ data. For example, In case of PDS data, information such as APL (Above poverty line), BPL (Below poverty line), Family Details, may be collected as part of KYR + data.

Following diagram depicts the relation between AADHAAR Enrolment Client and Registrar's KYR+ Application their integration.



The typical process to integrate the pre-enrolment data file with the AADHAAR Enrolment Client application is as follows:

- Resident comes to enrolment
- The enrolment operator complete the KYR data capture using AADHAAR Enrolment Client of the resident
- The KYR+ data as per the template gets saved into a .dat file (name-value pair) at a pre-defined location on the enrolment station
- The Resident's photograph gets saved into a .jpeg file) at a pre-defined location on the enrolment station
- The operator switches to the KYR+ application with the help of Alt+Tab key
- The KYR+data of the resident can be pulled to the KYR+ application with the help of a function key, for example, F5 key.
- KYR+ data is captured and saved by the KYR+ application

The KYR+ data gets captured from the AADHAAR client application after each enrolment capture and gets saved into a pre-defined location. At the end of every enrolment, a unique "Enrolment ID" is generated. This edited KYR data along with Enrolment ID is shared using a Name-Value pair file which will typically be a .dat file (kyrplus.dat) to Registrar's software for collecting KYR+ fields.

The KYR+ application requires almost the same set of data what AADHAAR system collects from the pre-enrolment data from registrar. The only extra column added is that of the Enrolment ID which is generated as a part of the enrolment from the application itself. The "Name" and the "Value" field will be separator with delimiter '=' (equal to) and name will carry the labels as mentioned below. For every new name-value pair, 'new Line' will be the separator, for example, FullName=Ram Prakash
DateOfBirth=10/04/1980

Mandatory	PreEnrolmentID
	LocalLanguageCode
Optional	EnrolmentClientVersion
	DocumentSpecificationVersion
	RegistrarCode
	EnrolmentAgencyCode
	StationID
	EnrolmentID
	ApplicationNumber
	NPR Receipt Number
	Bank Name
	Bank Branch Name
	Bank Branch IFSC Code
	Bank Account Number
	FullName
	FullName_LL
	DateOfBirth
	DateOfBirthType
	Gender
	AddrCareOf
	AddrCareOf_LL
	AddrBuilding
	AddrBuilding_LL
	AddrStreet
	AddrStreet_LL
	AddrLandmark
	AddrLandmark_LL
	AddrLocality
	AddrLocality_LL
	AddrVTC
	AddrVTC_LL
	AddrDistrict
	AddrDistrict_LL
	AddrState
	AddrState_LL
	PinCode
	RelativeName
	RelativeName_LL
	RelativeUID
	RelationType
	Mobile
	Email
	POType
	POName
	POName_LL
	DefaultPOName
	Verification Type

	POA
	POI
	Bank State

Note: The fields have been broken down to accommodate in this document.

The labels mentioned above are generated out of the AADHAAR Enrolment Client application. The description for all these labels are same as mentioned in the Pre-enrolment integration section. The labels suffixed with 'LL' are the respective details of the resident in the local language in which the AADHAAR Enrolment Client is configured for.

The **EnrolmentClientVersion** is the AADHAAR client application version number**.**

The **DocumentSpecificationVersion** is the registrar integration manual document version number.

The **RegistrarCode** is the 3 digit code of the Registrar.

The **EnrolmentAgencyCode** is the 4 digit code for the enrolment agency.

The **StationID** is the 5 digit code of the enrolment station.

The **EnrolmentID** label is the Enrolment Id of the resident. The Enrolment Id generated as a part of enrolment capture from the AADHAAR Enrolment Client application. This follow the data type format and length as described in section 3 of this document. The format of the Enrolment Id is as follows:
$N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}$YYYYMMDDHHMMSS where,

$$\left.\begin{array}{l} \bullet\ N_1N_2N_3N_4 \text{ – Enrolment Agency code} \\ \bullet\ \text{N5N6N7N8N9 – Station ID} \\ \bullet\ \text{N10N11N12N13N14- Sequence Number} \end{array}\right\} \text{Enrolment No.}$$

$$\left.\begin{array}{l} \bullet\ \text{YYYYMMDDHHMMSS – Date-timestamp} \end{array}\right\} \text{Enrolment Date in reverse order}$$

The **NPR Receipt Number** label is the National Population Register acknowledgement number of the resident. This shall adhere to the data type format and length as described in section 3 of this document.

The **PO Name** is the Post Office name of the resident. This shall adhere to the data type format and length as described in section 3 of this document. **PO Name_LL** is the local language equivalent of the same. **Default PO Name** is the default value of the Post Office name for the Pincode and Village/Town/City combination selected.

Verification Type captures the type of verification resident has provided. This can be of 3 types:

- Documents: Documents captures the Proof of Identity (PoI) and Proof of Address (PoA) of the residents

- Introducer: Introducer captures the Introducer details like Name and UID
- Head of Family (HoF): Head of Family captures the Head of the family details like Name, UID/EID, and type of relationship with the residents

Depending on type of verification, the respective details get populated in the KYRDAT file.

In case when the resident provides the age in years, the year of birth is calculated and passed on to the registrar specific application with the **Date of Birth** name value pair.

3. Data Type and Data Format

The data type and the data length for the various labels mentioned in the file format as mentioned in the previous sections are described below. The registrars are expected to follow the norms and provide the input file in the suggested data formats only.

Labels	Data Type	Data Length(max)
PreEnrolmentID	alpha-numeric (a-z, A-Z, 0-9) all special characters allowed	40
EnrolmentID	alpha-numeric (a-z, A-Z, 0-9, Space, colon, Hyphen, Front slash) • Enrolment Agency code (a-z, A-Z) • Station ID- (0-9) • Sequence Number (system generated 5 digit number) • Date Time Stamp (0-9,Hyphen,front slash) in YYYYMMDDHHMMSS format	40 [4] [5] [16]
LocalLanguageCode	numeric (0-9)	2
ApplicationNumber	alpha-numeric (a-z, A-Z, 0-9, Space, Hyphen)	40
TinNumber	alpha-numeric (a-z, A-Z, 0-9, Space, Hyphen)	15
NPR Receipt Number[2]	Same as TinNumber	Same as TinNumber
Bank State	Alphabet(a-z, A-Z, dot, Space)	40
Bank Name	Alphabet(a-z, A-Z, dot, Space)	40
Bank Branch Name	alpha-numeric (a-z, A-Z, 0-9, Space, Hyphen, Comma, parenthesis)	80
Bank Branch IFSC Code	alpha-numeric (a-z, A-Z, 0-9, Space, Hyphen, Comma, parenthesis)	20
Bank Account Number	alpha-numeric (a-z, A-Z, 0-9, Space, Hyphen)	20
FullName	alphabetic (a-z, A-Z, Space, dot)	99
FullName_LL	Indian language value of "Full Name"	99
DateOfBirth[3]	numeric (0-9)	8
DateOfBirthType[4]	alphabetic (a-z, A-Z)	15
Gender	char[M,F,T]	1
AddrCareOf	alpha-numeric (a-z, A-Z, 0-9, Space, # „	60

[2] The validations are same as that of Tin Number. Tin number and NPR receipt number are one and the same thing

[3] Date of Birth must be in DDMMYYYY format

[4] DOB type can be Approximate, Declared or Verified

	Comma, Dot, Front slash, Hyphen)	
AddrCareOf_LL	Indian language value of "AddrCareOf"	60
AddrBuilding	alpha-numeric (a-z, A-Z, 0-9, Space, # „ Comma, Dot, Front slash, Hyphen)	60
AddrBuilding_LL	Indian language value of "AddrBuilding"	60
AddrStreet	alpha-numeric (a-z, A-Z, 0-9, Space, # „ Comma, Dot, Front slash, Hyphen)	60
AddrStreet_LL	Indian language value of "AddrStreet"	60
AddrLandmark	alpha-numeric (a-z, A-Z, 0-9, Space, # „ Comma, Dot, Front slash, Hyphen)	60
AddrLandmark_LL	Indian language value of "AddrLandmark"	60
AddrLocality	alpha-numeric (a-z, A-Z, 0-9, Space, #„Comma, Dot, Front slash, Hyphen)	60
AddrLocality_LL	Indian language value of "AddrLocality"	60
AddrVTC	alpha-numeric (a-z, A-Z, 0-9, Space, # „ Comma, Dot, Front slash, Hyphen)	50
AddrVTC_LL	Indian language value of "AddrVTC"	50
AddrDistrict	alpha-numeric (a-z, A-Z, 0-9, Space, &)	50
AddrDistrict_LL	Indian language value of "AddrDistrict"	50
AddrState	alpha-numeric (a-z, A-Z, 0-9, Space, &)	50
AddrState_LL	Indian language value of "AddrState"	50
PinCode	numeric (0-9)	6
RelativeName	alphabetic (a-z, A-Z, Space, dot)	99
RelativeName_LL	Indian language value of "RelativeName"	99
RelativeUID	alpha-numeric (a-z, A-Z, 0-9, Space, Hyphen, Front slash)	30
RelationType	char[F,M,H,W,G]	1
Mobile[5]	numeric (0-9)	10
Email	alpha-numeric (a-z, A-Z, 0-9, Hyphen, Underscore, Dot, @)	254
POType	alphabetic (a-z, A-Z, Space, dot) example [B.O, S.O, H.O]	10
POName	alpha-numeric (a-z, A-Z, 0-9, Space, dot)	150
POName_LL	Indian language value of "POName"	150
DefaultPOName	alpha-numeric (a-z, A-Z, 0-9, Space, dot)	150
Verification Type	Alphabet (a-z, A-Z)	10
POA	alpha-numeric (a-z, A-Z, 0-9)	99
POI	alpha-numeric (a-z, A-Z, 0-9)	99
Introducer_Name	alphabetic (a-z, A-Z, Space, dot)	99
Introducer_UID	Numeric (0-9)	12
HOFProof	alpha-numeric (a-z, A-Z, 0-9)	99

[5] The mobile number shall not prefix '0' or +''

Registrars shall make sure that the string values for all the Villages/Districts/States names shall match exactly with the appropriate directories.

4. Enrolment Client XSDs

This section contains the schema design of the various files which are part of the enrolment client packet. The respective files have been embedded as an object in Appendix.

4.1 Demographic

The XSD contains field definition and structure of the resident demographics xml which is created when registrar enrolment packets are created. The xml will contain values from demographic details screens, static values from Face, Iris, Finger Print, and Review screen. The XSD can be used in application which will need to read the resident demographic details.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
targetNameSpace="http://www.uidai.gov.in/enrol/export/demographics/1.2.1"
  xmlns="http://www.uidai.gov.in/enrol/export/demographics/1.2.1"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
 <!--
    Root element for resident-demographics.xml file (normal capture)
  -->
 <xs:element name="uid-enrolment-profile" type="uid-enrolment-profile"/>
 <xs:complexType name="uid-enrolment-profile">
  <xs:all>
   <xs:element name="eid" type="xs:string" />
   <xs:element name="enrolment-number" type="xs:string" />
   <xs:element name="enrolment-date" type="xs:string" />
   <xs:element name="application-number" type="xs:string" minOccurs="0" />
   <xs:element name="pre-enrolment-id" type="xs:string" minOccurs="0" />
   <xs:element name="sspid" type="xs:string" minOccurs="0" />
   <xs:element name="jobcard-number" type="xs:string"
    minOccurs="0" />

   <xs:element name="res-name" type="xs:string" minOccurs="0" />
   <xs:element name="gender" type="GenderType" minOccurs="0" />
   <xs:element name="dob" type="DateOfBirthType" minOccurs="0" />
   <xs:element name="pincode" type="xs:string" minOccurs="0" />
   <xs:element name="careof" type="xs:string" minOccurs="0" />
   <xs:element name="building" type="xs:string" minOccurs="0" />
   <xs:element name="street" type="xs:string" minOccurs="0" />
   <xs:element name="landmark" type="xs:string" minOccurs="0" />
   <xs:element name="locality" type="xs:string" minOccurs="0" />

   <xs:element name="vtc" type="xs:string" minOccurs="0" />
   <xs:element name="vtc-name" type="xs:string" minOccurs="0" />
   <xs:element name="district" type="xs:string" minOccurs="0" />
   <xs:element name="district-name" type="xs:string" minOccurs="0" />
   <xs:element name="state" type="xs:string" minOccurs="0" />
   <xs:element name="state-name" type="xs:string" minOccurs="0" />

   <xs:element name="mobile" type="xs:string" minOccurs="0" />
   <xs:element name="email" type="xs:string" minOccurs="0" />
   <xs:element name="country" type="xs:string" minOccurs="0" />

   <xs:element name="lang-code" type="xs:string" minOccurs="0" />
   <xs:element name="local-res-name" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-careof" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-building" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-street" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-landmark" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-locality" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-vtc" type="xs:string" minOccurs="0" />
   <xs:element name="local-district" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-state" type="xs:string"
    minOccurs="0" />
   <xs:element name="local-country" type="xs:string"
    minOccurs="0" />

   <xs:element name="relation" type="RelationType"
    minOccurs="0" />

   <xs:element name="extended-attributes" type="ExtendedAttributes"
    minOccurs="0" />

   <xs:element name="biometric-capture" type="ResidentBiometricType"
    minOccurs="0" />

   <xs:element name="information-sharing-consent" type="xs:boolean" minOccurs="0"
/>
   <xs:element name="verification" type="VerificationProofType" minOccurs="0" />
   <xs:element name="mode" type="ModeType" />
   <xs:element  name="linked-enrolment"  type="LinkEnrolmentType"  minOccurs="0"
/>
   <xs:element name="financial-details" type="FinancialDetails" minOccurs="0" />
   <xs:element name="name-changed" type="xs:boolean" minOccurs="0" />
   <xs:element name="child-update" type="xs:boolean" minOccurs="0" />
  </xs:all>
  <xs:attribute name="version" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="VerificationProofType">
  <xs:choice>
    <xs:element name="Document" type="DocumentType"/>
    <xs:element name="Introducer" type="IntroducerType"/>
  </xs:choice>
</xs:complexType>


<xs:complexType name="DocumentType">
  <xs:sequence>
    <xs:element name="POI" minOccurs="0" >
      <xs:complexType>
        <xs:sequence>
          <xs:element name="documentType" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="POA" minOccurs="0" >
      <xs:complexType>
        <xs:sequence>
          <xs:element name="documentType" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="IntroducerType">
  <xs:sequence>
    <xs:element name="id" type="xs:string"/>
    <xs:element name="name" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="ModeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NEW" />
    <xs:enumeration value="UPDATE" />
    <xs:enumeration value="CORRECTION" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="LinkEnrolmentType">
  <xs:all>
```

```
    <xs:element name="link-id" type="xs:string"/>
    <xs:element name="link-id-type">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="UID" />
       <xs:enumeration value="EID" />
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
   </xs:all>
  </xs:complexType>
  <xs:complexType name="RelationType">
   <xs:all>
    <xs:element name="rel-id" type="xs:string" minOccurs="0" />
    <xs:element name="rel-id-type" minOccurs="0">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="UID" />
       <xs:enumeration value="EID" />
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
    <xs:element name="rel-name" type="xs:string" minOccurs="0" />
    <xs:element name="local-rel-name" type="xs:string"
          minOccurs="0" />
    <xs:element name="rel-type" minOccurs="0">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="F" />
       <xs:enumeration value="M" />
       <xs:enumeration value="H" />
       <xs:enumeration value="W" />
       <xs:enumeration value="G" />
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
   </xs:all>
  </xs:complexType>

  <xs:simpleType name="GenderType">
   <xs:restriction base="xs:string">
    <xs:enumeration value="M" />
    <xs:enumeration value="F" />
    <xs:enumeration value="T" />
   </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="DateOfBirthType">
   <xs:all>
```

```
    <xs:element name="day" type="xs:int" />
    <xs:element name="month" type="xs:int" />
    <xs:element name="year" type="xs:int" />
    <xs:element name="status">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="DECLARED" />
       <xs:enumeration value="APPROXIMATE" />
       <xs:enumeration value="VERIFIED" />
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
   </xs:all>
  </xs:complexType>


  <xs:complexType name="ExtendedAttributes">
   <xs:sequence>
    <xs:element name="Attribute" minOccurs="1" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="key" type="xs:string" />
       <xs:element name="value" type="xs:string" />
      </xs:sequence>
     </xs:complexType>
    </xs:element>

   </xs:sequence>
  </xs:complexType>

  <xs:complexType name="FinancialDetails">
   <xs:all>
    <xs:element          name="bankaccount-opening-consent"          type="xs:boolean"
minOccurs="0" />
    <xs:element name="state-name" type="xs:string" minOccurs="0" />
    <xs:element name="state-code" type="xs:string" minOccurs="0" />
    <xs:element name="bank-name" type="xs:string" minOccurs="0" />
    <xs:element name="bank-entity-code" type="xs:string" minOccurs="0" />
    <xs:element name="branch-name" type="xs:string" minOccurs="0" />
    <xs:element name="ifsc-code" type="xs:string" minOccurs="0" />
    <xs:element name="account-number" type="xs:string" minOccurs="0" />
   </xs:all>
  </xs:complexType>


  <!--
   Root element for operator-details.xml file
   -->
```

```
 <xs:element name="operator-details">
  <xs:complexType>
   <xs:complexContent>
    <xs:extension base="OperatorSupervisorType"></xs:extension>
   </xs:complexContent>
  </xs:complexType>
 </xs:element>

 <!--
  Root element for supervisor-details.xml file
  -->
 <xs:element name="supervisor-details">
  <xs:complexType>
   <xs:complexContent>
    <xs:extension base="OperatorSupervisorType"></xs:extension>
   </xs:complexContent>
  </xs:complexType>
 </xs:element>

 <xs:complexType name="OperatorSupervisorType">
  <xs:all>
   <xs:element name="opid" type="xs:string" minOccurs="0" />
   <xs:element name="name" type="xs:string" minOccurs="0" />
   <xs:element name="timestamp" type="xs:string" minOccurs="0" />
   <xs:element        name="biometric-capture"        type="BiometricCaptureType"
minOccurs="0" />
  </xs:all>
  <xs:attribute name="version" type="xs:string" use="optional" />
 </xs:complexType>

 <!--
  Root element for introducer-details.xml file
  -->

 <xs:element name="introducer-details" type="introducer-details"/>
 <xs:complexType name="introducer-details">
  <xs:all>
   <xs:element name="uid" type="xs:string" />
   <xs:element name="name" type="xs:string" />
   <xs:element name="timestamp" type="xs:string" minOccurs="0" />
   <xs:element name="biometric-capture" type="BiometricCaptureType"
        minOccurs="0" />
  </xs:all>
  <xs:attribute name="version" type="xs:string" use="optional" />
 </xs:complexType>

 <!--
    The resident's biometric data as a collection of capture trails for
    each group (left-slap, right-slap, face, left-iris, right-iris).
  -->

<xs:complexType name="ResidentBiometricType">
  <xs:sequence>
    <xs:element name="biometric-capture-trail" type="BiometricCaptureTrailType"
            minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<!--
     A collection of biometric captures corresponding to a specific group
     that can be left-slap, right-slap, face, left-iris, right-iris, etc.
  -->

<xs:complexType name="BiometricCaptureTrailType">
  <xs:sequence>
    <xs:element name="biometric-capture" type="BiometricCaptureType"
            minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="group" type="xs:string" use="required" />
</xs:complexType>

<!--
     Contains metadata information about a single biometric capture.
  -->

<xs:complexType name="BiometricCaptureType">
  <xs:all>
    <!--
            Device information as a collection of name-value pairs
        -->
    <xs:element name="device-info" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="parameter" type="DeviceParameterType"
                    minOccurs="1" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--
            List of body parts associated with this capture. For example, for a
            left slap, this would be left index, left ring, left pointer and
            left middle fingers.
        -->
    <xs:element name="body-part-list" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="body-part" type="BodyPartType"
                    maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--
            List of body parts missing from this capture (e.g. missing finger)
        -->
    <xs:element name="missing-part-list" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
            <xs:element name="body-part" type="BodyPartType"
                    maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

    <xs:element name="start-date" type="xs:string" minOccurs="0" />
    <xs:element name="end-date" type="xs:string" minOccurs="0" />
    <xs:element name="quality" type="xs:int" minOccurs="0" />
    <!--
            This should be the same as the index specified in the BIR info block
            in CBEFF XML.
        -->
    <xs:element name="cbeff-index" type="xs:string" minOccurs="1" />
    <xs:element name="forced" type="xs:boolean" minOccurs="0" />
    <xs:element name="best-attempt" type="xs:boolean"
            minOccurs="0" />

  </xs:all>
  <!--
      <xs:attribute name="start-date" type="xs:string" />
      <xs:attribute name="end-date" type="xs:string" />
      <xs:attribute name="quality" type="xs:int" />
      -->
  <!--
        This should be the same as the index specified in the BIR info block
        in CBEFF XML.
      -->
  <!--
      <xs:attribute name="cbeff-index" type="xs:string" />
      <xs:attribute name="forced" type="xs:boolean" />
      <xs:attribute name="best-attempt" type="xs:boolean" />
      -->
</xs:complexType>

<!--
      Contains information about a single device parameter in the form of a
      name-value pair.
  -->
```

```
<xs:complexType name="DeviceParameterType">
 <xs:sequence>
  <xs:element name="name" type="xs:string" />
  <xs:element name="value" type="xs:string" />
 </xs:sequence>
</xs:complexType>

<!--
 Contains information about a single body part. Type interpretation is as
 given below:

   Field Value Name | Biometric Type Value(hex)
   ==========================================
   No Information Given          000000
   Multiple Biometrics Used      000001
   Facial Features               000002
   Voice                         000004
   Fingerprint                   000008
   Iris                          000010
   Retina                        000020
   Hand Geometry                 000040
   Signature Dynamics            000080
   Keystroke Dynamics            000100
   Lip Movement                  000200
   Thermal Face Image            000400
   Thermal Hand Image            000800
   Gait                          001000
   Body Odor                     002000
   DNA                           004000
   Ear Shape                     008000
   Finger Geometry               010000
   Palm Print                    020000
   Vein Pattern                  040000
   Foot Print                    080000

 The biometric sub-type is arrived at using the following table for bitwise
 interpretation:

   B B B B B B B B
   8 7 6 5 4 3 2 1 Biometric Subtype
   ====================================
   0 0 0 0 0 0 0 0 No information given
         0 1 Right
         1 0 Left
     0 0 0   No meaning
     0 0 1   Thumb
     0 1 0   Pointer finger
     0 1 1   Middle Finger
     1 0 0   Ring Finger
```

```
        1 0 1     Little Finger
      x x x        Reserved for Future Use
  ex : 0 0 0 1 0 1 0 1 Right Little Finger
   -->

  <xs:complexType name="BodyPartType">
   <xs:attribute name="type" type="xs:int" />
   <xs:attribute name="sub-type" type="xs:int" />
  </xs:complexType>
</xs:schema>
```

4.2 Biometrics

The Biometrics XSD contains definition and structure in which biometric (Face, Iris, Finger Print, and Operator/ Supervisor/ Introducer authentication) details will be captured. The Registrar pack will contain resident-biometric, operator biometric, supervisor biometric and introducer biometric xml which will follow the below given structure. The XSD can be used to develop application which will read the packet details against the XSD.

The biometrics is stored in the JPEG 2000 format.

```
<?xml version='1.0' encoding="utf-8"?>
<xs:schema                              targetNameSpace="urn:oid:1.1.19785.0.257.1.7.0"
xmlns="urn:oid:1.1.19785.0.257.1.7.0"
          xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
         attributeFormDefault="unqualified">

  <xs:element name="bir" type="bir" />

  <xs:complexType name="bir">
   <xs:sequence>
    <xs:element name="version" type="RevisionType" minOccurs="0" />
    <xs:element name="cbeff-version" type="RevisionType" minOccurs="0" />

    <xs:element name="bir-info" type="BIRInfoType" minOccurs="0" />
    <xs:element name="bdb-info" type="BDBInfoType" minOccurs="0" />
    <xs:choice>
     <xs:element name="bir" type="bir" minOccurs="0"
                                        maxOccurs="unbounded" />
    </xs:choice>
    <xs:choice id="optBdb">
     <xs:element name="bdb" type="xs:base64Binary" minOccurs="0" />
    </xs:choice>
   </xs:sequence>
  </xs:complexType>

  <xs:complexType name="RevisionType">
```

```
   <xs:attribute name="major" type="xs:int" />
   <xs:attribute name="minor" type="xs:int" />
  </xs:complexType>
  <xs:complexType name="BIRInfoType">
   <xs:sequence>
    <xs:element name="creator" type="xs:string" minOccurs="0" />
    <xs:element name="index" type="xs:string" minOccurs="0" />
   </xs:sequence>
   <xs:attribute name="integrity" type="xs:boolean" use="optional" />
   <xs:attribute name="creation-date" type="xs:string" use="optional" />
   <xs:attribute name="not-valid-before" type="xs:string"
                     use="optional" />
   <xs:attribute name="not-valid-after" type="xs:string"
                     use="optional" />
  </xs:complexType>

  <xs:complexType name="BDBInfoType">
   <xs:attribute name="format-owner" type="xs:string" />
   <xs:attribute name="format-type" type="xs:string" />
   <xs:attribute name="encryption" type="xs:boolean" use="optional" />
   <xs:attribute name="creation-date" type="xs:string" />
   <xs:attribute name="type" type="xs:string" use="optional" />
   <xs:attribute name="subtype" type="xs:string" use="optional" />
   <xs:attribute name="level" type="xs:string" use="optional" />
   <xs:attribute name="product-owner" type="xs:string" use="optional" />
   <xs:attribute name="product-type" type="xs:string" use="optional" />
   <xs:attribute name="purpose" type="xs:string" use="optional" />
   <xs:attribute name="quality" type="xs:string" use="optional" />
  </xs:complexType>
</xs:schema>
```

4.3 Registrar Report (EID-UID mapping)

The XSD will produce a registrar report in XML file format which will be sent to registrars informing about the status of the enrolments done by them. The registrar report is generated for every registrar and contains information about accepted as well as rejected enrolments and this process is run every day.

```
<?xml version='1.0' encoding="utf-8"?>
<xs:schema                    targetNameSpace="urn:oid:1.1.19785.0.257.1.7.0"
xmlns="urn:oid:1.1.19785.0.257.1.7.0"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">

<xs:element name="bir" type="bir" />

<xs:complexType name="bir">
```

```
        <xs:sequence>
                <xs:element name="version" type="RevisionType" minOccurs="0"
/>
                <xs:element name="cbeff-version" type="RevisionType"
minOccurs="0" />

                <xs:element name="bir-info" type="BIRInfoType" minOccurs="0"
/>
                <xs:element name="bdb-info" type="BDBInfoType"
minOccurs="0" />
                <xs:choice>
                        <xs:element name="bir" type="bir" minOccurs="0"
                                maxOccurs="unbounded" />
                </xs:choice>
                <xs:choice id="optBdb">
                        <xs:element name="bdb" type="xs:base64Binary"
minOccurs="0" />
                </xs:choice>
        </xs:sequence>
</xs:complexType>

<xs:complexType name="RevisionType">
        <xs:attribute name="major" type="xs:int" />
        <xs:attribute name="minor" type="xs:int" />
</xs:complexType>
<xs:complexType name="BIRInfoType">
        <xs:sequence>
                <xs:element name="creator" type="xs:string" minOccurs="0" />
                <xs:element name="index" type="xs:string" minOccurs="0" />
        </xs:sequence>
        <xs:attribute name="integrity" type="xs:boolean" use="optional" />
        <xs:attribute name="creation-date" type="xs:string" use="optional" />
        <xs:attribute name="not-valid-before" type="xs:string"
                use="optional" />
        <xs:attribute name="not-valid-after" type="xs:string"
                use="optional" />
</xs:complexType>

<xs:complexType name="BDBInfoType">
        <xs:attribute name="format-owner" type="xs:string" />
        <xs:attribute name="format-type" type="xs:string" />
        <xs:attribute name="encryption" type="xs:boolean" use="optional" />
        <xs:attribute name="creation-date" type="xs:string" />
        <xs:attribute name="type" type="xs:string" use="optional" />
        <xs:attribute name="subtype" type="xs:string" use="optional" />
        <xs:attribute name="level" type="xs:string" use="optional" />
        <xs:attribute name="product-owner" type="xs:string" use="optional" />
        <xs:attribute name="product-type" type="xs:string" use="optional" />
        <xs:attribute name="purpose" type="xs:string" use="optional" />
        <xs:attribute name="quality" type="xs:string" use="optional" />
</xs:complexType>
</xs:schema>
```

5. Appendix

demographics_v1.2.
1.xsd

biometrics.xsd

registrarreport.xsd